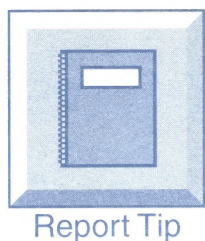




# Inside MICROSOFT ACCESS

September 1993 • Vol. 1 No. 7

Tips & techniques for Microsoft Access • Windows



## Creating bar code labels in Access reports



**Y**ou see bar codes everywhere these days. The grocery store uses bar codes to record the prices of the items you buy; the post office uses them to track mail during transit.

The big advantages of using bar codes are speed and accuracy. The light pen swipes across the bar code very quickly and rarely misreads the code. Have you ever been at the supermarket checkout counter when the scanner doesn't read a product's code properly? The person at the register must then type in the product's code. Fingers on the keypad often make mistakes the first time, and even if they're correct, fingers don't enter code nearly as fast as the scanner.

Unfortunately, the appearance of bar codes gives most people the impression that bar codes are too complex for small companies or individuals to use. However, the concept behind bar codes is very simple. And with Access on your side, bar code labels are easy to create. In this article, we'll show you a simple method for producing bar code labels you can stick on your inventory items.

### What are bar codes?

Most people don't realize that bar codes are simply coded values. Each character in the entry you want to encode has a corresponding sequence of lines and spaces in the bar code format. When you encode the full entry, each character in the value contributes its lines and spaces to the full bar code.

Understanding that, you can see how bar codes might play a role in your database. Most tables have a primary key—even if it's only a counter field. All you need to do is print out labels, expressing those ID entries as bar codes. Then you can stick the labels on the object the record identifies.

At the supermarket, the bar codes identify an item so that the register can look up the product's record in a database and type its name and price on the receipt.

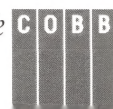
### Creating bar code labels on an Access report

So, how can you create bar code labels with Access? Well, you must obtain a bar code font that you can install in Windows along with the other fonts.

After you install a bar code font under Windows, you can generate bar code labels

### IN THIS ISSUE

- Creating bar code labels in Access reports ..... 1
- Downloading files from CompuServe ..... 5
- Using Windows 3.1's Character Map to paste special characters on your forms ..... 6
- Combo box controls let you display a field other than the bound field ..... 8
- Using a query as the row source of a combo box control ... 10
- Don't give a query the same name as a table—or vice versa ..... 13
- Microsoft includes runtime version in its Access Distribution Kit ..... 14
- Printing a report for a single record ..... 14





by using that font in an Access report. You simply place a text box control to print the ID entry and then assign the bar code font to that control. When you print the report, the bar codes will appear rather than the usual numbers and letters.

### 3-of-9 bar codes

One popular type of bar code is the 3-of-9 system. In our example, we'll use a TrueType font—called RSCoDe39—that uses this system. We downloaded the font from the Access libraries on CompuServe. It's freeware, so you can download it yourself or copy it from the Inside Microsoft Access Resource Disk. See the article on page 5 for more information on downloading the font file.

### Installing the new fonts

Once you download the font files, you must install them in the Windows envi-

ronment. You do so by using the Windows Control Panel application. You'll find its icon (🖨️) in the Program Manager's Main group. Double-click the icon to launch the application. Then, double-click the Fonts icon (🔤) in the Control Panel window. Next, click the Add... button in the Fonts window and specify the fonts you want to add in the resulting Add Fonts dialog box.


To add a font, first look in the Directories list for the directory that holds the font files. Then, highlight the font file (or files when you install more than one font from that directory) and click OK. After you complete these steps and restart Windows, the font name will appear in the font list in all your Windows applications.

### Using the new fonts in an Access report

After you've installed the RSCoDe39 font, launch Access so you can create a new report that prints data in this font. We'll use the supermarket example we've been discussing all along. (However, note that supermarkets typically use UPC bar codes rather than the 3-of-9 system.)

Open your test database and, while the Database window lists the tables, click the New button. In the new Table window,

Table A

Supermarket Products Table			
Key	Field Name	Data Type	Field Properties
	Product ID	Text	Field Size = 8
	Product Name	Text	
	Short Name	Text	
	Apply Tax	Yes/No	
	Price	Currency	

## Inside MICROSOFT ACCESS™

Inside Microsoft Access (ISSN 1067-8204) is published monthly by The Cobb Group.

**Prices:** Domestic ..... \$59/yr. (\$7.00 each)  
Outside US ..... \$79/yr. (\$8.50 each)

**Phone:** Toll free ..... (800) 223-8720  
Local ..... (502) 491-1900  
Customer Relations Fax ..... (502) 491-8050  
Editorial Department Fax ..... (502) 491-4200

**Address:** You may address tips, special requests, and other correspondence to

The Editor, *Inside Microsoft Access*  
9420 Bunsen Parkway, Suite 300  
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to

Customer Relations  
9420 Bunsen Parkway, Suite 300  
Louisville, KY 40220

**Postmaster:** Second class postage is pending in Louisville, KY. Send address changes to  
*Inside Microsoft Access*  
P.O. Box 35160  
Louisville, KY 40232

#### Copyright:

Copyright © 1993, The Cobb Group. All rights reserved. *Inside Microsoft Access* is an independently produced publication of The Cobb Group. The Cobb Group reserves the right, with respect to submissions, to revise, republish, and authorize its readers to use the tips submitted for both personal and commercial use.

The Cobb Group, its logo, and the Satisfaction Guaranteed statement and seal are registered trademarks of The Cobb Group. *Inside Microsoft Access* is a trademark of The Cobb Group. Paradox is a registered trademark of Borland International. dBASE III and dBASE III PLUS are registered trademarks of Ashton-Tate, a Borland International company. Microsoft and MS-DOS are registered trademarks of Microsoft Corporation. Microsoft Windows and Word for Windows are trademarks of Microsoft Corporation.

#### Staff:

Editor-in-Chief ..... David Brown  
Editing ..... Timothy E. Hampton  
Elizabeth Welch  
Production Artist ..... Julie Jefferson  
Design ..... Karl Feige  
Publications Manager ..... Tara Dickerson  
Managing Editor ..... Suzanne Thornberry  
Circulation Manager ..... Brent Shean  
Publications Director ..... Linda Baughman  
Editorial Director ..... Jeff Yocom  
Publishers ..... Mark Crane  
Jon Pyles

#### Back Issues:

To order back issues, call Customer Relations at (800) 223-8720. Back issues cost \$7 each, \$8.50 outside the US. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all articles that have appeared in *Inside Microsoft Access*.



enter the table structure shown in Table A. Make sure you define the Product ID field as the table's primary key. You do so while designing the table by clicking its row selector and then clicking the Primary Key button (P) on the toolbar. The key symbol will appear next to the field in Design view.

Then, pull down the File menu and select the Save As... command. Next, type *Supermarket Products* and click OK. Finally, click the Datasheet button (D) on the tool bar, enter the sample data shown in Figure A, and then close the window.

Now, highlight the Supermarket Products table when the table list appears and then click the New Report button (N) on the tool bar. When the New Report dialog box appears, click the ReportWizards button. In the following dialog box, select Mailing Label and click OK.

With the first wizard dialog box, you lay out the fields on the label. Double-click the Product Name field first. Then, click the Return button (R). Next, select the Price field and click the Return button. Finally, include the Product ID field. Figure B shows the final layout.

Then, click the Next button at the bottom of the dialog box. The window that appears asks you to specify the fields the report should sort by. Accept the default by clicking the Next button. The dialog box that follows asks for the Avery number of the label sheets you want to print on. In this example, we'll use number 4162, which creates 15/16" x 3 1/2" labels. Figure C shows that the 4162 entry is the seventh item in the list. Click Next to continue. In the wizard's last dialog box, click the Design button.

The report wizard will generate the report shown in Figure D. As you can see, the wizard has placed text box controls for the fields and has sized the Detail section to the proper size of the label. The wizard also sets the

Figure A

Table: Supermarket Products					
Product ID	Product Name	Short Name	Apply Tax	Price	
LK1897J	Mustard	Must	No	\$1.59	
LK345KJ	Mayonnaise	Mayo	No	\$2.79	
LK12940U	Ketchup	Ketc	No	\$2.29	
LKU308I	Tabasco	Tabasc	No	\$3.59	
*				\$0.00	

We'll use this sample data to demonstrate our bar code printing technique.

Figure B

You lay out the fields with the first dialog box the Mailing Label wizard presents.

Figure C

Select your label's Avery number with the Mailing Label wizard.

Figure D

The Mailing Label wizard generates this report.



margins and other aspects of the report layout, which you can see by selecting the Print Setup... command from the File menu. We won't discuss the layout details here.

You'll need to make a few adjustments to the default report. Click the Properties button (🔧) on the tool bar to open the property sheet. Then, click the Price field's control. In the property sheet, scroll to the end of the property list and change the Text Align property from *Left* to *Right*. Next, click on the Product ID field's control and assign *Right* to its Text Align property as well.

Now make the Product ID field's control display the Product ID entries as bar codes by changing the Font Name property to *RSCode39*. Also, you'll need to increase the Font Size property from the default 8 points to 24 points. When you do, you'll notice the bar codes don't fit in the control. Stretch the control by assigning *0.4* to its Height property. Figure E shows the resulting report.

### Note the RSCode39 font's allowable characters

When you use the RSCode39 font, you must make sure the code contains only uppercase letters and numbers. The font doesn't supply bar code characters for the lowercase letters or punctuation characters. If you try, blank space will appear in the bar code.

Figure F shows the character map for the RSCode39 font. You can examine the font by using the Character Map application (which, incidentally, we used to produce Figure F). For more information on using the Character Map, see "Using Windows 3.1's Character Map to Paste Special Characters on Your Forms," on page 6.

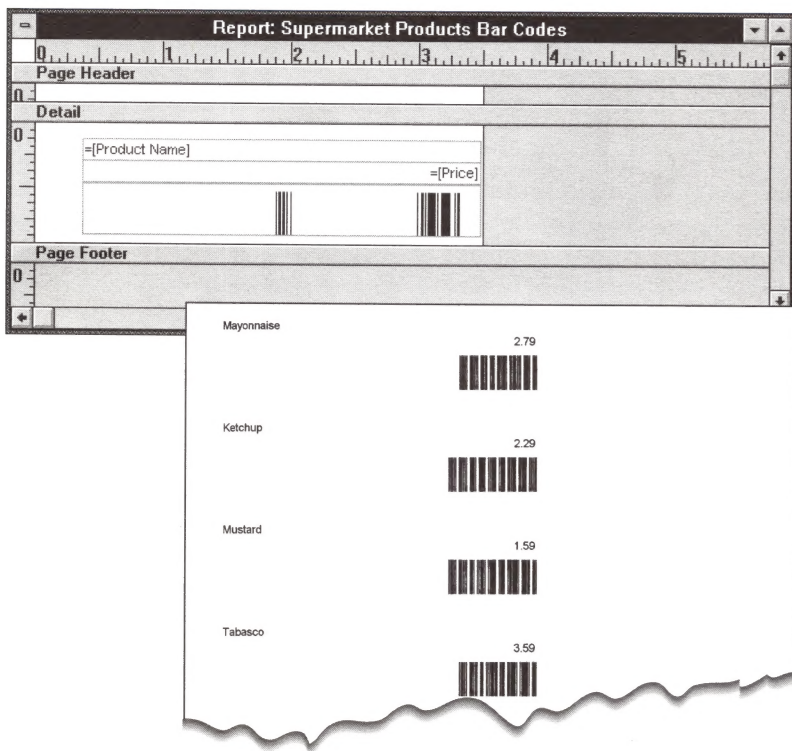
### Reading bar code labels

Creating the bar code labels is only half the story: You must also be able to read the bar codes from the labels. The supermarket buys sophisticated scanners that supply the Product ID entries to the cash registers. You might need to buy only a light pen or some other piece of less-sophisticated hardware. Such equipment becomes another input device—an alternative to the keyboard and mouse. Swiping the pen over the code is equivalent to typing the entry. The only difference is the bar-code-reading equipment is faster and doesn't make data entry mistakes.

### Conclusion

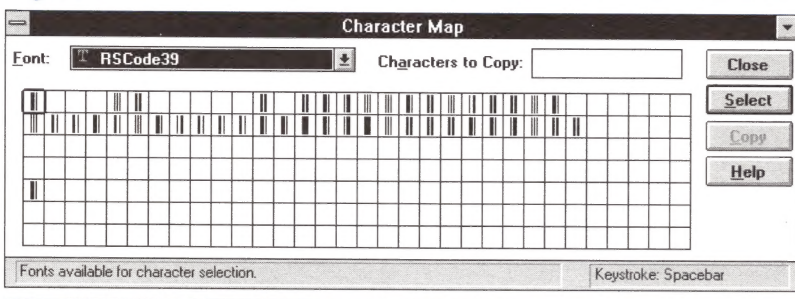
Bar codes have many uses. You'll need to consider how you want to track items in order to determine how bar codes will help you. Once you define the role of bar codes in your database, the technique we've shown you will help you produce the bar code labels. ♦

Figure E



After modifying a few properties, the report will produce bar codes as shown here.

Figure F



Here's the character map for the RSCode39 font.



**ORDER TODAY!**

## Is This Your Copy?

If Not, Order Your Subscription Today!



Enter my subscription for one year of *Inside Microsoft Access* for only \$49. That's just \$4.08 a copy instead of the usual \$6 an issue!

**Please bill me later.**

Name \_\_\_\_\_

Firm \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone ( \_\_\_\_\_ ) \_\_\_\_\_ S 2066

Outside U.S.: \$8.50 ea., \$69/yr.

VCJ1

*Inside*  
**MICROSOFT ACCESS**



**NEW ORDER!**



NO POSTAGE  
NECESSARY  
IF MAILED IN  
THE UNITED  
STATES

**BUSINESS REPLY MAIL**

FIRST CLASS MAIL

PERMIT NO. 618

LOUISVILLE, KENTUCKY

POSTAGE WILL BE PAID BY ADDRESSEE



THE COBB GROUP  
PO BOX 35160  
LOUISVILLE KY 40232-9719





# Downloading files from CompuServe

**Y**ou may run into some problems as you download the bar code font from the CompuServe library. In this short article, we'll briefly explain the downloading process. We'll also describe the problems many people are having these days with the PKZIP and PKUNZIP utilities.

## Downloading files

We'll first walk you through the steps to download a file in an Access CompuServe library. We'll use as an example the file CODE39.ZIP, which contains the RSCODE39 font we used in the accompanying article.

After logging on, enter *GO MSACCESS* at the ! prompt. This command will move you directly to the Access forum. After the forum menu scrolls by, enter 3 to select the 3 *LIBRARIES* option. Next, CompuServe lists the 16 forum libraries you have to choose from. You select the library you want by entering the number next to its name. For instance, to download the font file, enter 15 to select the 3rd Party/User Grp library.

Another menu will then appear, which describes the things you can do in the library. You can select 1 *BROWSE*, 2 *DIRECTORY*, 3 *UPLOAD*, 4 *DOWNLOAD*, or 5 *LIBRARIES*. To download a file, you enter 4. CompuServe will prompt you first for the name of the file you want to download from the library and then for the protocol you want to use for the file transfer. Finally, you must issue the command in your telecommunication software that receives the file, matching the protocol you chose in issuing CompuServe's *DOWNLOAD* command.

You can use the above instructions because you know the name of the file you want to download—CODE39.ZIP. However, if you don't know the filename, you must use the *BROWSE* or *DIRECTORY* option on the menu to determine the filename. The *BROWSE* option lets you provide keywords for searching the library's files. *DIRECTORY* simply lists every file in the library.

## What's happening with the PKZIP and PKUNZIP utilities?

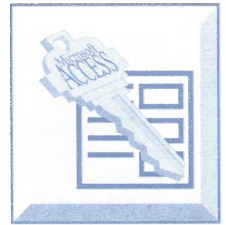
As you probably know, almost every file in any CompuServe library is compressed by

using the PKZIP utility. After you download a file, you must decompress it by using the PKUNZIP utility. If you plan to upload and download files, you should have both PKZIP and PKUNZIP. Obtaining these utilities is easy—especially when you use the Access forum on CompuServe. The utilities are in the 3rd Party/User Grp library—the same library the font resides in.

Although downloading PKZIP and PKUNZIP is very easy, setting up the utilities is a little confusing. Their developers have recently produced a new version that isn't backward compatible with the previous version. In other words, files you zip with the upgraded PKZIP can't be unzipped with the previous version of PKUNZIP.

As a result of this dilemma, we suggest you download both the PKZ110.EXE and PK204G.EXE files—with the latest upgrades for each version—so you'll be ready for anything. We routinely compress files by using Version 1.1 so that the person receiving the compressed file will be able to unzip it with either version. We decompress files with Version 2.04g, since that utility can handle files compressed with either version.

These files are self-extracting. In other words, after you download them, you can simply run them, and they'll decompress themselves into the files PKZIP.EXE and PKUNZIP.EXE and many others. Also, note that these utilities are shareware. The files you download contain an order form that has all the information you need to register your copy. ♦

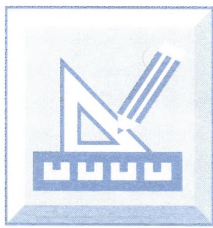


Access Tip

## A correction

We'd like to point out an error in the June article "Get to Know the Microsoft Knowledge Base." In that article, we suggested you download the Microsoft Knowledge Base from CompuServe so it would always be available on your computer. Unfortunately, the library name in which the file resides changed after we went to press with that issue. To obtain the Knowledge Base, download the file ACCKB.ZIP from Library 1, Getting Started.





Design Tip

# Using Windows 3.1's Character Map to paste special characters on your forms



When you type at your computer, you usually use only the 100 or so alphanumeric and punctuation characters. However, all the fonts that ship with Windows 3.1 include 255 characters. For the most part, the additional characters are symbols or other special characters that don't have their own keys—usually they require an awkward combination of keystrokes to produce them. In this article, we'll show you how to use the Windows 3.1 Character Map application to view those special characters and then place them on your Access forms and reports.

## Viewing all the characters in a character set

Windows 3.1 includes an application called Character Map in the Accessories group. You can use this application to view all the characters in a font's character set. After you launch the application by double-clicking its icon (📁) in Program Manager, the Character Map will appear, as shown in Figure A. At that point, you can select a font and Windows will display its character set on the 32-column, 7-row grid. In Figure A, the application shows the characters in the Arial font.

Once you've found a character you want to place on your form, you can copy it to the Clipboard, then paste it in your Access forms and reports. You can paste the character anywhere you can type ordinary characters. As an alternative, you can read the key combination at the bottom-right corner of the window in the status bar, then return to Access and type the character.

By the way, if you click on a character in the map and hold down the mouse button,

the Character Map application enlarges the character so you can see it more clearly. This feature comes in very handy when you're finding a character in the Symbol or Wingdings font.

## Placing a character on an Access form or report

You often want to place the character or characters in a label control while designing a form or report. Well, we have good news and bad news about pasting characters.

The good news is Access automatically creates a label control to hold the incoming character when you issue the Paste command. You can then position the control as appropriate.

The bad news is the control Access creates has the label control's default properties. Most importantly, the characters will appear in the control's default font rather than the font you were viewing in the Character Map. If you haven't changed the defaults, all label controls initially have a Font Type property setting of *MS Sans Serif*. Also, they have a default Font Size property setting of 8.

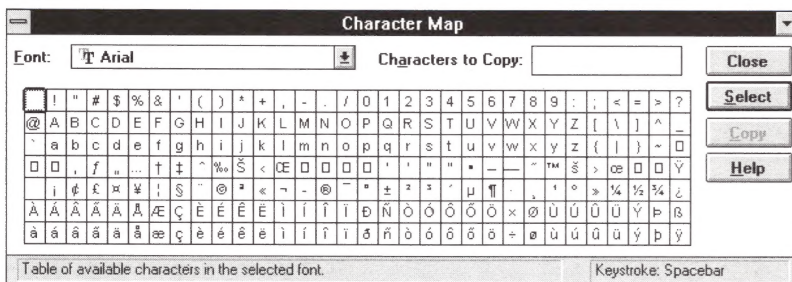
Consequently, copying characters from the Character Map application and pasting them into an Access form or report is easy. However, you must open the property sheet and change the control's Font Type property if you need to display or print the character in a specific font. For example, the G in the Wingdings font is not even a letter—it's a hand with its index finger pointing up (☞).

## An example

Suppose you want to place the registered trademark symbol (®) next to the product name *Widget Mark IV* in the report shown in Figure B. Unfortunately, you don't know the keystrokes to produce the symbol. Therefore, you use the Character Map to locate the character and paste it in your document.

First, you'll need to copy the symbol from the Character Map dialog box to the Clipboard. To do so, open the Character Map application and then click on the registered trademark symbol, as shown in Figure C. Most fonts define the symbol as the character 0174, which places it in the fifth row of

Figure A



These are all the characters in the Arial font's character set.



the Character Map's grid—below *N* and *n*.

Note that we created the screen shot in Figure C while holding down the mouse button on the registered trademark symbol. As we mentioned, holding down the mouse button on a character tells the Character Map application to display an enlarged view of the character. As soon as you release the mouse button, the character's cell in the grid returns to normal.

After you highlight the symbol, click the Select button. Windows will place the symbol in the Characters To Copy text box, as shown in Figure D. Then, copy the symbol to the Clipboard by clicking the Copy button.

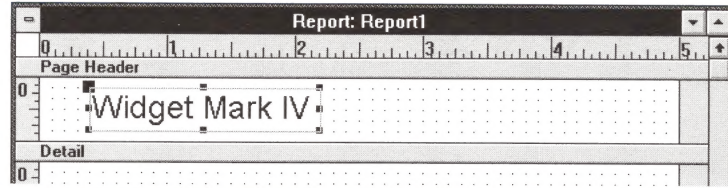
Now that the symbol is in the Clipboard, you return to Access and the report in which you want to paste the symbol. Select the Page Header section, where the *Widget Mark IV* label resides, by clicking in an empty area of the section. Then, pull down the Edit menu and select Paste. Access will create a new label control in the section's upper-left corner, as shown in Figure E.

Next, you grab the label control and move it to the right of the *Widget Mark IV* label. Then, you can open the property sheet and change the Font Name property to the font that matches the label's font. Figure F shows the result.

## Conclusion

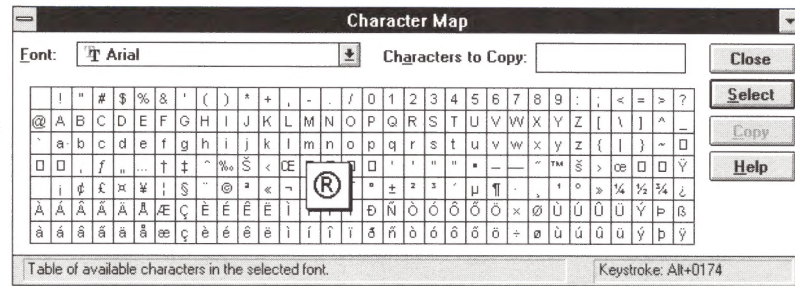
In this article, we showed you how to use the Character Map application that comes with Windows 3.1 in order to include symbols and other special characters in your forms and reports. You can either copy the characters to the Clipboard and then paste them directly in a form or report, or you can use the Character Map to determine the key combination and then type the character directly. If you use the Clipboard to paste the character in a form or report, Access will automatically create a label control to contain the character. ♦

Figure B



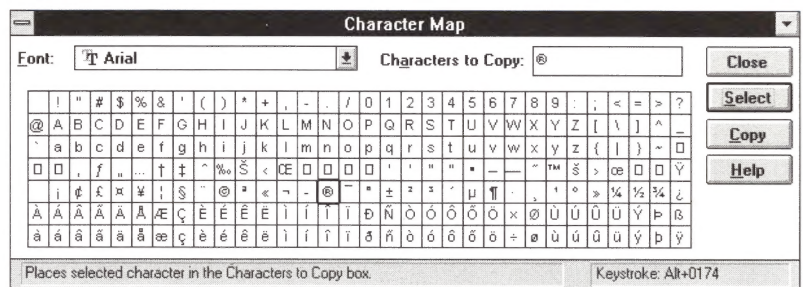
You want to enter the registered trademark symbol next to the *Widget Mark IV* string, but you don't know the key combination to type it.

Figure C



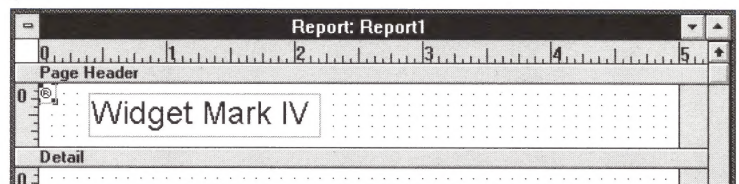
You can highlight the character you want to use by pointing to it and clicking your mouse, or you can enlarge the character by holding the mouse button after you click.

Figure D



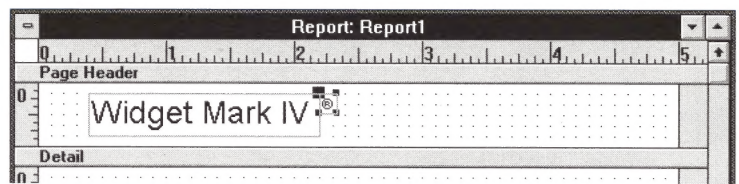
You can tell Windows you want to use the registered trademark symbol by clicking the Select button.

Figure E



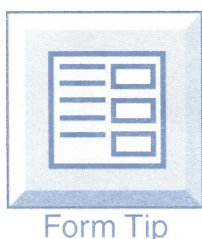
You can paste the registered trademark symbol in the report section by using the Edit menu's Paste command.

Figure F



As you can see, the button's name includes the registered trademark symbol.





Form Tip

## Combo box controls let you display a field other than the bound field



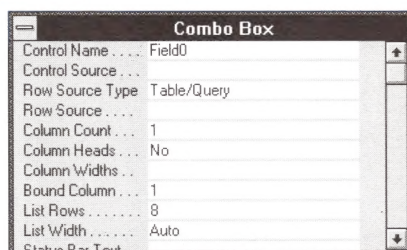
**H**ave you ever used a combo box control to access another table from a form? Since the control's selection list can display several fields of the other table, the combo box provides an excellent lookup-help facility. For instance, suppose you want to keep track of the sales representatives who close sales with customers. To do so, you include a Sales Rep ID field in the Orders table. Then, on the order-entry form, you use a combo box that displays the Sales Reps table's data in its selection list. That way, you could choose a Sales Rep ID by reviewing the sales reps' name data.

In this article, we'll first run through a primer for creating such a basic combo box. Then, we'll show you how to enhance the combo box by selecting the best field to display in the combo box control. In other words, we'll show you how to configure the combo box so it displays the name data but continues to store the ID values. By using this technique, you'll be able to work with descriptive data during data entry, even though the table stores the primary key entries necessary to maintain the relationship between the tables.

### Defining a combo box control that accesses tables

Let's first discuss the basic technique for accessing tables with a combo box. We'll start by examining the combo box's properties. Figure A shows the first several properties of a default combo box.

Figure A



You configure a combo box by setting the properties shown here.

The first two properties should be familiar to you. Most controls have Control Name

and Control Source properties. The Control Name property contains the name you can use to refer to the control in expressions. The Control Source property specifies the table or query field to which you bind the control. In the order-form example we suggested, you want to name the combo box Orders' Sales Rep ID and then bind the control to the Orders table's Sales Rep ID field. To do so, you'd assign *Orders' Sales Rep ID* to the Control Name property and *Sales Rep ID* to the Control Source property.

The next several properties define the combo box's selection list. The Row Source Type property specifies the type of data the control will list. As you can see from the figure, the property's default is *Table/Query*, which is the setting you want in order to list data from a lookup table. Then, the Row Source property defines the actual table or query that provides the data for the list.

Next, you can indicate how many of the table's or query's fields you want to display in the combo box's selection list by using the Column Count property. For instance, if you enter 3, the list will contain the table's first three fields. The Column Heads property that follows lets you display the names of the Row Source's fields at the top of the selection list.

The Column Widths property lets you define the width of each column in inches. If you include more than one field in the list, you separate each column's width with a semicolon. You use this property to hide a field: You simply specify 0 as the field's column width. Also, if you want to let Access apply the default column width, just omit an entry for that column. For example, if a combo box has three fields and you want to hide the first one, accept the default width for the second column, and set the third column's width to 2 inches, you'd assign to the Column Widths property the string *0 in;;2 in*.

Next, the Bound Column property lets you choose which column to match with the control's bound field you entered in the Control Source property. In the order-entry example, you use the default value of 1 because, as the primary key field, Sales Rep ID is the first

field in the Sales Reps table. Remember, you want to select the column that matches the Sales Rep ID field in the Orders table you specified in the Control Source property.

Finally, the List Rows and List Width properties define the dimensions of the selection list. List Rows specifies the number of rows that will appear in the list. List Width defines the width in inches. You can enter Auto in the List Width property in order to set the list width to the width of the control.

## An example

Let's look at a quick example. Suppose you need to keep track of your company's sales. You define an Orders table to keep track of sales data such as the order date and transaction amount. Table A shows the structure of the fields you might include.

As you can see, the table includes the Number field Sales Rep ID. This field lets you keep track of the employees at your company who close sales with customers.

During data entry, you might want some help in entering the correct ID value for a particular sales rep. To do so, you can create another table to keep track of your company's sales reps, and you can reference the table in a combo box on the order-entry form. Table B shows the structure of the Sales Reps table, and Figure B shows some sample data.

Figure B

Table: Sales Reps	
Sales Rep ID	Sales Rep Name
1	David Brown
2	Bob Thompson
3	James Artner
4	Alice Jones
5	Suzanne Smith
(Counter)	

The Sales Reps table holds basic information about your company's sales reps.

## Creating a simple order-entry form

Let's design a form for the Orders table that uses a combo box for the Sales Rep ID entry. First, you'll create a basic form with the Single-column FormWizard. Then you'll replace the form's default Sales Rep ID text box with a combo box.

Start by highlighting the Orders table in the database window. Then, click the New Form button (📄) on the tool bar. In the New

Table A

Orders Table			
Key	Field Name	Data Type	Field Properties
🔑	Order ID	Counter	
	Order Date	Date/Time	
	Sales Rep ID	Number	Field Size =Long Integer
	Customer ID	Number	Field Size =Long Integer
	Sales Amount	Currency	

Table B

Sales Reps Table			
Key	Field Name	Data Type	Field Properties
🔑	Sales Rep ID	Counter	
	Sales Rep Name	Text	Field Size = 50

Figure C

The Single-column FormWizard creates this report for the Orders table.

Form dialog box, click FormWizards and, in the resulting dialog box, select the Single-column item and click OK. When you do, Access will produce the first of the wizard's dialog boxes. For this example, just click the Fast Forward button (➤) in the lower-right corner, which tells Access to select all default options from the dialog boxes and move directly to the last dialog box. Then, click the Design button. Access will generate the form shown in Figure C.

## Replace the Sales Rep ID field's text box with a combo box

Next, you'll replace the Sales Rep ID text box control with a combo box that will list the reps in the Sales Reps table. Start by deleting the Sales Rep ID text box. Just highlight the control and press [Del].



Before you create the combo box, make sure the Toolbox and field list are open. To open the Toolbox, select its option on the

View menu; to display the field list, click the Field List button (📋) on the tool bar.

Now you're ready to create the combo box for the Sales Rep ID. You do so by first selecting the Combo Box tool (📋) in the Toolbox; then, click the Sales Rep ID field in the field list and drag the field to the empty space below the Order Date field. When you release the mouse button, a default combo box will appear. Before continuing, position and size the control to match the other fields, as shown in Figure D.

Figure D

Place a bound combo box for the Sales Rep ID field.

### Displaying the Sales Rep Name field in the combo box

When you're satisfied with the combo box's location, open the property sheet by clicking the Property Sheet button (📋) on the tool bar. You'll see the *Sales Rep ID* entry, which is already stored in the Control Name and Control Source properties.

## Using a query as the row source of a combo box control

In the accompanying article, we show you how to use a combo box to access the data from another table on a data-entry form. The selection list will offer the allowable selections for the field. However, you often don't want to include all the records in the table. Instead, you want to limit the list to a smaller set of data. In this article, we'll modify the other article's example to demonstrate how you'd use a query to customize the list of entries a combo box provides.

### Replacing the Sales Reps table with the Employees table

Our original example uses a table named Sales Reps as the combo box's row source. However, you may not want to use this table. Instead, you might use an Employees table, such as the one shown in Table A. This table stores all of a company's employee data, including a Job Type field that categorizes workers by their job function. Figure A shows some sample data.

On the order-entry form, you need a combo box that lists your sales staff so you can easily match sales representatives with the Orders records. To provide the items in the list, you create a query that displays the Employee ID and Name fields of only the workers that have the entry *Sales* in the Job Type field.

To create such a query, highlight the Employees table in the Database window and click the New Query button (📋) on the tool

Table A

Employees Table			
Key	Field Name	Data Type	Field Properties
📋	Employee ID	Counter	
	Name	Text	Field Size = 50
	Job Type	Text	Field Size = 20

Now you're ready to configure the combo box's selection list. Move to the Row Source property, click the dropdown arrow, and select the Sales Reps table from the list. Next, assign the number 2 to the Column Count property. Doing so tells Access to include the Sales Reps table's first two fields—Sales Rep ID and Sales Rep Name—in the control's selection list. Then, assign 1 to the Bound Column property in order to match the two tables' Sales Rep ID fields.

### Using the new combo box

In a moment, we'll modify the combo box to display the Sales Rep Name field in the combo box. But first, let's switch to Form view to see how the combo box operates at this stage. Click the Form View button (📄) on the tool bar. When you click the Sales Rep ID combo box's dropdown arrow, the selection list will display all available Sales Rep ID entries in the Sales Reps table, as shown in Figure E.

Figure E

The combo box we've created lists the Sales Rep ID entries.

Since we defined two columns for the combo box, you can view the Sales Rep Name as you choose a sales rep from the

bar. When the Query window appears, drag the Employee ID, Name, and Job Type fields from the field list to the QBE grid. Next, deselect the Job Type column's Show check box. Finally, move to the column's Criteria cell and type "Sales". Figure B shows the resulting query and its Datasheet view that you see when you click the Datasheet button (📄) on the tool bar.

Before you can incorporate this query in the order-entry form's combo box, you must save it. Pull down the File menu and select Save Query As.... Then, enter *Sales Reps From Employees* and click OK.

Figure A

Employee ID	Name	Job Type
1	David Brown	Sales
2	Jeff Zing	Manager
3	Bob Thompson	Sales
4	Kelly Walther	Accounting
5	James Artner	Sales
6	Sonya Smith	Accounting
7	Alice Jones	Sales
8	Everett Boing	Programmer
9	Suzanne Smith	Sales
	(Counter)	

We'll use this sample data in our example.

Now, while you design the order-entry form, you can open the property sheet for the Sales Rep ID combo box and assign the new query to the Row Source property rather than to a separate Sales Reps table.

Figure B

This query selects your sales staff's employee records.



selection list. After pulling down the selection list, you click the horizontal scroll bar's right arrow button. The selection list will show the second column, which contains the Sales Rep Name entries, as shown in Figure F.

Figure F

You can also view the selection list's second column, which displays the Sales Rep Name entries.

## Choosing the combo box's display field

Although the combo box we just created is helpful in selecting a Sales Rep ID during data entry, we can improve it. Instead of displaying the Sales Rep ID field, the combo box can display the Sales Rep Name field. When you configure the combo box to display that field, you don't need to think in terms of the cryptic ID field. You can just select or type the sales rep's name.

To modify the combo box, return to Design view by clicking the Design View button (🔍) on the tool bar. Then, move to the Column Widths property in the property sheet and enter 0 in;1 in, as shown in Figure G. This setting will hide the Sales Rep ID field and allow 1 inch for the Sales Rep Name field. Also, it

tells Access to display the Sales Rep Name field entry from the row you select in the combo box control's text box.

## Using the modified combo box

Now let's test the completed combo box control. Click the Form View button again to

open the form for data entry. Next, click the combo box's dropdown arrow to see the Sales Reps table's data. Figure H shows the form as you select a sales rep. (Note that we've resized the window to fit the form by using the Size To Fit Form command on the Window menu.)

Figure H

The combo box's selection list offers the Sales Rep Name entries.

When you select an entry, the combo box control will store the Sales Rep Name value rather than the Sales Rep ID. Figure I shows the form after we've selected the *James Artner* entry. However, keep in mind that the underlying Orders table is storing the Sales Rep ID entry associated with the rep's name.

Figure I

When you select a Sales Rep Name, Access displays the name even though the control is bound to the Sales Rep ID field.

If you want to verify that Access stores the Sales Rep ID, you can open the Orders table's datasheet and take a look. But first, you must save the new record by selecting Save Record from the File menu. Next, return to the Database window by pressing [F11]. Then, click the Tables button and double-click the Orders entry. Figure J shows the Orders datasheet that appears. As you can see, the Sales Rep ID field contains the number 3, which is James Artner's Sales Rep ID entry.

As we mentioned, you can even use the keyboard to provide an entry. For instance, if you want to enter *David Brown* into the field,

Figure G

After you configure the combo box, the property sheet should look like this.

you can type the name rather than use the selection list. When you do, the combo box will display the name in the control, but the combo box will still store the corresponding Sales Rep ID value, 1, in the Orders table.

You can even type a partial entry that uniquely defines your selection. For example, you could type only *David*—or even *D*—and then press [Enter]. (There are no other entries that begin with the letter *D*.) Access will automatically fill in *David Brown*.

## Conclusion

In this article, we provided a detailed explanation of how to use combo box controls to provide lookup help on a form. The combo box can list another table's data in order to help you select the entry you want.

Figure J

Table: Orders				
Order ID	Order Date	Sales Rep ID	Customer ID	Sales Amount
1		3		\$0.00
(Counter)		0	0	\$0.00

The Orders table's datasheet will show the actual value the combo box stores—in this instance, 3.

We also showed you how to select the field to display in the combo box in the text box part of the control. This technique is especially helpful when the field to which you bind the control is Counter or another type of ID field. You can work with more descriptive fields during data entry while the form actually stores the ID field in the table. ♦

## Don't give a query the same name as a table—or vice versa

If you've worked with Access for long, you know you can base forms and reports on queries as well as on tables. You can even base queries on other queries. In fact, you might as well consider queries as tables, since Access does just that in almost every instance.

The constant blending of table and query names has an important consequence: You can't give a table and a query the same name. When you try to give a query a name that a table is already using (or vice versa), Access will pop up a dialog box that warns you of the conflict. However, the message in the dialog box is a little confusing, and if you aren't careful, your response to the dialog box can delete the table or query.

Essentially, Access pops up the same dialog box it would if you had duplicated the name of a like object: It will ask if you want to replace the existing object. For instance, if you enter an existing table name while renaming a query, Access will ask you to confirm that you want to overwrite the table. Also, if you try to give a table the name of an existing query, Access will ask if you want to replace the existing query.

Of course, you'd *never* want to do such a thing. You wouldn't want to replace a table

that stores data with a query that only looks up data. As a result, the warning message can be very confusing.

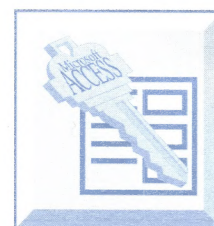
Furthermore, if you click the Yes button, which tells Access to replace the object, Access does so. If you accidentally duplicate a table's name while naming a query, you'll lose the table and all the data in it.

## Remember the trap!

The only way to avoid deleting a table or query is simply to avoid duplicate names. You'll probably never run into this problem if you always use long, descriptive names when you name tables and queries. In fact, the situation is most likely to arise when you're in a hurry and use short names. For instance, when you're researching an Access feature or otherwise trying to make your database more efficient, you're less likely to pay attention to simple tasks such as naming queries.

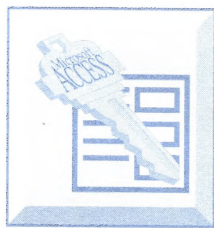
## Notes

This naming conflict exists only between tables and queries. For instance, Access lets you give a form the same name as a query or give a report the name of an existing form. ♦



Access Tip





### Access Tip

## Microsoft includes runtime version in its Access Distribution Kit

In the second quarter, Microsoft began shipping the Access Distribution Kit (ADK). The kit includes several utilities that let you package an Access database as a standalone application. The ADK contains a utility with which you create an icon for your database application and a help compiler that converts any text you've written in a word processor, such as Word for Windows, to a Windows Help application.

The most important piece of software in the ADK is Access Runtime, which allows developers to package and distribute their Access applications without requiring users to purchase the full Access product. Runtime allows unlimited distribution with Access applications.

The ADK also comes with a fairly concise booklet outlining how to create and distribute your applications. This booklet also includes a chapter on how to use Access security in order to protect your code.

Runtime requires the same amount of memory (RAM) as the full version of Access and runs in the same environment. The only difference is that the application a developer distributes by using Runtime may require less disk space, since the computer doesn't

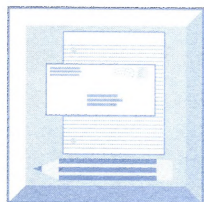
need to store the sample applications, Cue Cards, and other peripheral files.

Runtime offers a couple of advantages over the full Access version. First, you can hide the application's Database window. Also, Runtime doesn't display the Microsoft Access splash screen on startup. Those two changes remove your application's visible connection to Access.

The other big difference is that Runtime doesn't offer Design view for any type of object. It simply launches an application whose Autoexec macro opens forms and reports. The user can then interact with the forms and reports by using the capabilities you defined when you developed the application.

You usually won't have to change existing applications for Runtime. Even if your application makes ad hoc changes to objects, the application should work as it would in the full product without modification—as long as the application makes changes by using DoMenuItem and SendKeys actions. The documentation will outline any other changes you might want to make.

Call Microsoft's Product Support line to order the ADK. The phone number is (800) 426-9400. ♦



### Letters

## Printing a report for a single record

How do you select and print one row of an Access report so that the printout includes the column headings and Report Header? I can't find any help with this problem from the documentation or any of the books I've bought.

As the editor of a newsletter—and the keeper of the mailing list—I often need to excerpt one subscriber's name and address information and print it out for the subscriber to verify. I can do it easily in dBASE III PLUS, but I can't seem to find a way in Access. Can you help?

*C.B. Caldwell  
Alpin, California*

We have two answers to Mr. Caldwell's question. First, if you'd be content seeing the data in a form's printout, the familiar Print... command on the File menu provides an easy solution. On the other hand, if you need to use a report to print the record, the solution is more complex.

We'll look at both techniques, using Mr. Caldwell's situation as an example. Suppose you use a table named Subscribers to store your newsletter's subscriber information. You want to view this data with a form called Subscriber List and, while viewing the form, print a report named Subscriber Info for individual subscribers. Figure A shows the table's





datasheet, the form, and the Design view of the report.

## Printing the record from a form

Let's start with printing the record by using a form. In this situation, you're viewing the Subscriber List form, and you want to produce hard copy of a certain subscriber's name and address. You first click the row selector for the record. Then, you pull down the File menu and select the Print... command. In the Print dialog box, click the Selection option button in the Print Range section and then click OK. Access will print the form showing only the data in the row you selected.

In most cases, printing from a form is the easiest method. Even if you want the look of a particular report, you can often create a form that looks close enough to suffice. Then, you can find the record you want to print while you view the form. Now, you can just print the record as we described.

## Printing the record from a report

Sometimes, the situation demands that you use a report to print the data. You want to find the record by using the form and then print the report for only that record. In this section, we'll show you a macro that does the job. You can then place a button on the form that executes the macro. Once you place the button on the form, you can isolate the record you want to print and then simply click the button.

The macro you need contains the OpenReport action. As you may know, this action has a Where argument, which lets you define the records you want to print. In this case, you use the Where argument to limit the report to printing only the current record. After you build this macro, you place a button on the form that executes the macro. Then, you can view the form, find the record you want to print, and click the button.

It's worth pointing out that you must use a form rather than the table's datasheet to implement this technique. Why? Well, one obvious answer is that you must place the button on *something*. However, there's another, more important reason: To formulate the criteria you put in the Where argument, you must be able to refer to the current

Figure A

Subscribers Table:

Table: Subscribers						
Subscriber ID	Subscriber Name	Address	City	State	ZIP Code	
1	Adele Williams	401 Rodeo Dr.	Louisville	KY	40220	
2	Alan Buttersworth	7775 S.W. Clinton Ave.	Louisville	KY	40220	
3	Alan Thompson	7384 Washington Ave.	Louisville	KY	40220	
4	Anton Dvorak	7 Nachoes Way	Louisville	KY	40220	
5	April Cienkewicz	4242 Maple Blvd.	Louisville	KY	40220	
6	Art Braunschweiger	P.O. Box 555	Louisville	KY	40220	
7	Bill Lee	418 Datablitz Ave.	Louisville	KY	40220	
8	Chiara Santorini	45 N. Terminal Way	Louisville	KY	40220	
9	Fran Wilson	89 Chiaroscuro Rd.	Louisville	KY	40220	
10	George Eddington	Franklin Mall	Louisville	KY	40220	
11	Gladys Lindsay	213 E. Roy St.	Louisville	KY	40220	
12	Hank Prouditt	45 E. 23rd St.	Louisville	KY	40220	
(Counter)						

Subscriber List Form:

Subscriber List

Subscriber ID	Subscriber Name	Address
1	Adele Williams	401 Rodeo Dr. Louisville KY 40220
2	Alan Buttersworth	7775 S.W. Clinton Ave. Louisville KY 40220
3	Alan Thompson	7384 Washington Ave. Louisville KY 40220

Subscriber Info Report:

Report: Subscriber Info

Report Header

Subscriber Info

=Now()

Page Header

Detail

Subscriber ID: Subscriber ID

Subscriber Name: Subscriber Name

Address: Address

City State ZIP Code

Page Footer

We'll use these objects in describing our printing techniques.

record's field entries. The only way to do that in Access is to use identifiers in expressions that return the values in the form's controls.

## Building the macro

Now, let's create the macro. In the Database window, view the list of macros and click the New button. When the Macro window appears, the cursor will reside in the first row's Action cell. Click the dropdown arrow and choose OpenReport from the selection list.



**Microsoft Access**  
**Technical Support**  
 (206) 635-7050

Please include account number from label with any correspondence.

Then, move to the Action Arguments section to provide the details about the report you want to open. First, enter *Subscriber Info* in the Report Name argument. Next, change the View argument from the default *Print Preview* to *Print*. Finally, move to the Where argument and enter

```
[Subscriber ID] =  
➔ [Forms]![Subscriber List]![Subscriber ID]
```

where the [Subscriber ID] on the left of the equal-to sign refers to the Subscriber table's primary key field and the [Subscriber ID] on the right refers to the control on the form that displays the primary key field.

When you enter this expression into the Where argument, Access will print only those entries that have a Subscriber ID entry that matches the one shown in the form control. Of course, this record is always the current record.

Now, save the macro. Pull down the File menu and select Save As.... When the Save As dialog box appears, type *Print Current Using Subscriber Info* and click OK. Then, close the Macro window.

Next, you place the button on the form. Open the Subscriber List form in Design view by double-right-clicking its entry in the Database window. Then, press [F11] to return to the Database window and click the Macro button. Find the Print Current Using Subscriber Info macro and drag it to the Subscriber List form's Form Footer section. A button will appear that will execute the macro. The button will have the full macro name as its caption. To shorten the caption, click within the button to get the insertion point cursor. Then, remove the Using Subscriber Info portion of the macro name and press [Enter]. Figure B shows the final form.

Now, you're ready to test the form's new button. Just click the Form View button (🖨️) on the tool bar to view the data through the form. Then, decide on a record you want to print and click the Print Current button. ♦

Figure B

Subscriber ID	Subscriber Name	Address
1	Adele Williams	401 Rodeo Dr. Louisville KY 40220
2	Alan Buttersworth	7775 S.W. Clinton Ave. Louisville KY 40220
3	Alan Thompson	7384 Washington Ave. Louisville KY 40220

Print Current

Record: 1

The Print Current button on the Subscriber List form prints single-record reports.

## Subscribe to the Inside Microsoft Access Resource Disk!



Do you wish you could experiment with the forms, reports, tables, macros, modules, and queries we regularly feature in *Inside Microsoft Access* but don't have the time or patience to create them? If so, you may want to subscribe to the Inside Microsoft Access Resource Disk. Once you subscribe, we'll send you a monthly disk loaded with all the useful tips featured in that month's *Inside Microsoft Access*.

A six-month subscription to the Inside Microsoft Access Resource Disk costs \$29. A full one-year subscription is \$49. If you don't want to subscribe but would like the forms, reports, tables, macros, modules, and queries in a particular issue of *Inside Microsoft Access*, you can purchase the disk for only \$9.95.

To subscribe or order a specific month's disk, just call our Customer Relations department at (800) 223-8720. Outside the US, please call (502) 491-1900.

